IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Dewitt, Jr. et al.**	§	
	§	Group Art Unit: **2191**
Serial No. **10/675,872**	§	
	§	Examiner: **Vo, Ted T.**
Filed: **September 30, 2003**	§	
	§	
For: **Method and Apparatus for Counting Instruction and Memory Location Ranges**	§	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on January 4, 2007.

A fee of $500.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-25

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: NONE.
2. Claims withdrawn from consideration but not canceled: NONE.
3. Claims pending: 1-25.
4. Claims allowed: NONE.
5. Claims rejected: 1-25.
6. Claims objected to: NONE.

C. CLAIMS ON APPEAL

The claims on appeal are: 1-25

STATUS OF AMENDMENTS

An Amendment after Final Office Action was not filed.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a method in a data processing system for monitoring execution of instructions. An instruction for execution is identified (**3200, Figure 32**; page 62, lines 3-4). It is determined whether the instruction is within a contiguous range of instructions (**3202, Figure 32**; page 63, lines 4-6), and execution information relating to the instruction is generated if the instruction is within the contiguous range of instructions (**3208, Figure 32**; page 63, lines 21-23).

B. CLAIM 8 - INDEPENDENT

The subject matter of claim 8 is directed to a method in a data processing system for monitoring access to data in memory location An access to data in a memory location is identified (**3200, Figure 32**; page 62, lines 3-4 and page 63, lines 24-27). It is determined whether the memory location is within a contiguous range of memory locations (**3202, Figure 32**; page 63, lines 4-6 and page 63, lines 24-27), and information relating to the memory location is generated if the memory location is within the contiguous range of memory locations (**3208, Figure 32**; page 63, lines 21-23 and page 63, lines 24-27).

C. CLAIM 15 – INDEPENDENT

The subject matter of claim 15 is directed to a data processing system for monitoring execution of instructions. The data processing system includes identifying means for identifying an instruction for execution (**214, 218, Figure 2**; page 16, lines 24-27), and determining means for determining whether the instruction is within a contiguous range of instructions (**3106, 3108, Figure 31**; page 61, line 29-page 62, line 3). Generating means generate execution information relating to the instruction if the instruction is within the contiguous range of instructions (**3110, Figure 31**; **240, Figure 2**; page 62, lines 7-15).

D. CLAIM 22 – INDEPENDENT

The subject matter of claim 22 is directed to a data processing system for monitoring access to data in memory locations. The data processing system includes identifying means for identifying an access to data in a memory location (**216, Figure 2**; page 16, lines 4-16), and determining means for determining whether the memory location is within a contiguous range of memory locations (**216, Figure 2**; page 62, lines 16-24). A generating means generates information relating to the memory location if the memory location is within the contiguous range of memory locations (**3110, Figure 31**; **240, Figure 2**; page 62, lines 7-24).

E. CLAIM 24 – INDEPENDENT

The subject matter of claim 24 is directed to a computer program product in a computer readable medium for monitoring execution of instructions. The computer program product includes first instructions for identifying an instruction for execution (**3200, Figure 32**; page 62, lines 3-4), and second instructions for determining whether the instruction is within a contiguous range of instructions (**3202, Figure 32**; page 63, lines 4-6). Third instructions are provided for generating execution information relating to the instruction if the instruction is within the contiguous range of instructions (**3208, Figure 32**; page 63, lines 21-23).

F. CLAIM 25 - INDEPENDENT

The subject matter of claim 25 is directed to a computer program product in a computer readable medium for monitoring access to data in memory locations. The computer program product includes first instructions for identifying an access to data in a memory location (**3200, Figure 32**; page 62, lines 3-4 and page 63, lines 24-27), and second instructions for determining whether the memory location is within a contiguous range of memory locations (**3202, Figure 32**; page 63, lines 4-6 and page 63, lines 24-27). Third instructions are provided for generating

information relating to the memory location if the memory location is within the contiguous range of memory locations (**3208**, **Figure 32**; page 63, lines 21-23 and page 63, lines 24-27).

G. CLAIM 4 - DEPENDENT

The subject matter of claim 4, which depends from claim 1, recites that the method further includes determining whether the instruction is within a second contiguous range of instructions (**3204**, **Figure 32**; page 63, lines 10-13), and generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions (**3208**, **Figure 32**; page 63, lines 21-23).

H. CLAIM 6 - DEPENDENT

The subject matter of claim 6, which depends from claim 1, specifies that the determining step includes comparing an address of the instruction to a set of addresses in a set of registers in a processor to determine whether the instruction is in the contiguous range of instructions (**3210**, **Figure 32**; page 63, lines 4-7).

I. CLAIM 11 - DEPENDENT

The subject matter of claim 11, which depends from claim 8, recites that the method further includes determining whether the memory location is within a second contiguous range of memory locations (**3204**, **Figure 32**; page 63, lines 10-13, page 63, lines 24-27), and generating the information relating to the memory location if the instruction is within the second contiguous range of memory locations (**3208**, **Figure 32**; page 63, lines 21-23, page 63, lines 24-27).

J. CLAIM 13 – DEPENDENT

The subject matter of claim 13, which depends from claim 8, specifies that the determining step includes comparing an address of the memory location to a set of addresses in a set of registers in a processor to determine whether the memory location is in the contiguous range of memory locations (**3210**, **Figure 32**; page 63, lines 4-7).

K. CLAIM 18 – DEPENDENT

The subject matter of claim 18, which depends from claim 15, specifies that the determining means is a first determining means and the generating means is a first generating means, and that the system further includes second determining means for determining whether the instruction is within a second contiguous range of instructions (**3106**, **3108**, **Figure 31**; page 61, line 29-page 62, line 3), and second generating means for generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions (**3110**, **Figure 31**; **240**, **Figure 2**; page 62, lines 7-15).

L. CLAIM 20 – DEPENDENT

The subject matter of claim 20, which depends from claim 15, specifies that the determining means comprises comparing means for comparing an address of the instruction to a set of addresses in a set of registers in a processor to determine whether the instruction is in the contiguous range of instructions (**3106**, **3108**, **Figure 31**; page 61, line 29-page 62, line 3).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION 1 (Claims 24-25)

Claims 24-25 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

B. GROUND OF REJECTION 2 (Claims 1-25)

Claims 1-25 are rejected under 35 U.S.C. § 102(b) as being anticipated by "A Hardware-Driven Profiling Scheme for Identifying Program Hot spot to Support runtime Optimization" to Merten et al.

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 24-25)

Claims 24-25 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

The Examiner asserts that claims 24 and 25 fail to satisfy the requirements of 35 U.S.C. § 101 because the specification defines a computer readable medium as including "transmission-type media such as digital and analog communication links" which the Examiner contends is non-statutory. This rejection is respectfully traversed.

Appellants submit that no basis is present for holding a computer usable medium claim non-statutory because the medium may be allegedly "intangible." The MPEP states:

> In this context, "functional descriptive material" consists of **data structures** and computer programs **which impart functionality when employed as a computer component.** (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993). "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data.
>
> **When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized.** Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium that increases computer efficiency held statutory) and *Warmerdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim) with *Warmerdam*, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure *per se* held nonstatutory). **(Emphasis added.)**

MPEP 2106 (IV)(B)(1)

Claims 24-25 recite clearly functional descriptive material since they impart functionality when employed as a computer component. Moreover, the functional descriptive material of claims 24-25 is recorded on "some" computer-readable medium.

In the above context, the term "some" means "any" computer-readable medium. The

MPEP does not draw any distinctions between one type of media that is considered to be statutory and another type of media that is considered to be non-statutory. To the contrary, the MPEP clearly states that as long as the functional descriptive material is in "some" computer-readable medium, it should be considered statutory. The only exceptions to this statement in the MPEP are functional descriptive material that does not generate a useful, concrete, and tangible result, e.g., functional descriptive material composed completely of pure mathematical concepts that provide no practical result. Claims 24-25 clearly recite a useful, concrete and tangible result. For example, claim 24 specifies instructions for generating execution information relating to the instruction if the instruction is within the contiguous range of instructions, and claim 25 specifies instructions for generating information relating to the memory location if the memory location is within the contiguous range of memory locations. Claims 24 and 25 do not recite some disembodied mathematical concept or abstract idea.

Thus, claims 24-25 are directed to functional descriptive material that provides a useful, concrete and tangible result, and which is embodied on "some" computer-readable medium. Therefore, claims 24-25 are statutory and it is respectfully requested that the Board reverse the Examiner's Final Rejection of those claims.

B. GROUND OF REJECTION 2 (Claims 1-25)

Claims 1-25 are rejected under 35 U.S.C. § 102(b) as being anticipated by "A Hardware-Driven Profiling Scheme for Identifying Program Hot spot to Support runtime Optimization" to Merten et al.

B.1 Claims 1-7, 15-21, and 24

In finally rejecting the claims, the Examiner states:

> As per Claim 1: Merten discloses,
> *A method in a data processing system for monitoring execution of instructions, the method comprising:*
> *identifying an instruction for execution;*
> *determining whether the instruction is within a contiguous range of instructions*
> (In run-time of a program, Merten depicts a code block: See left col., p. 137:2-18; depicts a code region: See left col., sec. 2, p. 138: These read limitation, "*contiguous range of instructions*". For execution, when the execution falls in the determined code region, it is known that every executed instruction in the region is within, and this is the purpose in which Merten runs the code for monitoring and identifying hotspots);

and generating execution information relating to the instruction if the instruction is within the contiguous range of instructions (Merten uses profiling to identify *execution information*, and to particularly identify the executions of branches within that code region: See sec. 2.1.1, p. 139).

Final Office Action dated November 3, 2006, page 5.

Claim 1 of the present application is as follows:

> 1. A method in a data processing system for monitoring execution of instructions, the method comprising:
>
> identifying an instruction for execution;
>
> determining whether the instruction is within a contiguous range of instructions; and
>
> generating execution information relating to the instruction if the instruction is within the contiguous range of instructions.

A prior art reference anticipates a claimed invention under 35 U.S.C. § 102 only if every element of the claimed invention is identically shown in that single prior art reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of a claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983).

Appellants respectfully submit that Merten does not identically show every element of the claimed invention arranged as they are in the claims, and, accordingly, does not anticipate the claims. With respect to claim 1, in particular, Appellants respectfully submit that Merten does not teach or suggest the claimed steps of "determining whether the instruction is within a contiguous range of instructions", or "generating execution information relating to the instruction if the instruction is within the contiguous range of instructions".

Merten is directed to a mechanism for identifying program "hot spots" to support runtime optimization. In rejecting the claims as being anticipated by Merten, the Examiner refers particularly to left column, page 137, lines 2-18 and left column, section 2, page 138 of Merten, reproduced below for the convenience of the Board:

> We have observed that many applications exhibit behavior conducive to runtime profiling and optimization. For example, program execution often occurs in distinct phases, where each phase consists of a set of code blocks that are executed with a high degree of temporal locality. When a collection of intensively executed blocks also has a small static footprint, it represents a highly favorable opportunity for runtime optimization. We will refer to such sets of blocks and their corresponding periods of execution as *hot spots*. A runtime optimizer can take advantage of execution phases by isolating a group of hot spots that are active for each phase. Ideally, aggressive optimized code would be deployed early in the phase and the optimized code used until execution shifts to another phase. Optimized hot spots that are no longer active may then be discarded, if necessary, to reclaim memory space for newly optimized code.
>
> Our proposed hot spot detection scheme uses three criteria to classify a region of code as a hot spot. First, the region must have a small static code size to facilitate rapid optimization. Second, the hot spot must be active over a certain minimum time interval so that it is likely to have an opportunity to benefit from runtime optimization. Finally, the instructions in the selected region of code must account for a large majority of the dynamic execution during its active time interval. These three criteria are sufficient to detect code regions that can benefit most from runtime optimization without placing unnecessary restrictions on the type of hot spots that can be identified.

Nowhere in the above recitations or elsewhere in Merten is it described that a determination is ever made whether an instruction that is identified for execution is within a contiguous range of instructions. Instead, Merten describes sets of code blocks that are executed with a high degree of temporal locality ("hot spots"), and describes three criteria by which a region of code is specified as a hot spot "that can benefit most from runtime optimization". As indicated above, the criteria include having a small static code size, being active over some time interval, and accounting for a large majority of dynamic execution during its active time interval. Merten does not describe identifying an instruction for execution, and certainly does not disclose or suggest determining whether such identified instruction is within a contiguous range of instructions. Merten does not make a determination whether an instruction identified for execution is within a contiguous range of instructions, but, instead, selects a set of code blocks based on the three criteria described above.

In responding to Appellants' arguments in this regard, the Examiner asserts that those skilled in the art know that one block of instructions is a contiguous range of instructions, and the act of picking an instruction in a block of instructions is an act of "determining" (Final Office

Action dated November 3, 2006, page 3). Appellants respectfully disagree. Initially, Merten nowhere defines what is meant by a set of code blocks. Certainly, Merten does not disclose that a set of code blocks is a contiguous range of instructions. The Examiner appears to be reading subject matter into Merten that is simply not disclosed in the reference.

Furthermore, picking an instruction is not at all the same as determining whether a particular instruction that has been identified for execution is within a contiguous range of instructions, and, in fact, appears to be quite the opposite of determining whether a particular instruction that has been identified for execution is within a contiguous range of instructions.

For similar reasons, Merten also does not disclose or suggest "generating execution information relating to the instruction if the instruction is within the contiguous range of instructions" as recited in claim 1. Again, Merten selects a set of code blocks (which are nowhere disclosed as comprising a contiguous range of instructions) based on specified criteria, but does not generate execution information relating to an instruction that is identified for execution, and that has been determined to be within a contiguous range of instructions. Merten, instead, identifies a set of code blocks "that can benefit most from runtime optimization" based on the criteria specified in Merten.

Claim 1, accordingly, is not anticipated by Merten and patentably distinguishes over Merten in its present form.

Independent claims 15 and 24 recite similar subject matter as claim 1, and are also not anticipated by Merten for similar reasons as discussed above with respect to claim 1.

Claims 2-7 depend from and further restrict claim 1, and claims 16-21 depend from and further restrict claim 15. These claims are also not anticipated by Merten at least by virtue of their dependency.

B.2 Claims 8-14, 22-23, and 25

Independent claim 8 is as follows:

> 8. A method in a data processing system for monitoring access to data in memory locations, the method comprising:
> identifying an access to data in a memory location;

determining whether the memory location is within a contiguous range of memory locations; and

generating information relating to the memory location if the memory location is within the contiguous range of memory locations.

Merten nowhere discloses or suggests "determining whether the memory location" to which an access to data has been identified "is within a contiguous range of memory locations" as recited in claim 8, nor has the Examiner identified any such disclosure in Merten. Similarly, Merten does not disclose or suggest "generating information relating to the memory location if the memory location is within the contiguous range of memory locations" as also recited in claim 8.

In rejecting claim 8, the Examiner states that it is rejected on the same rationale as claim 1. Claim 8, however, recites different subject matter than claim 1. Merten does not disclose monitoring memory location accesses, and does not anticipate claim 8 for this reason as well as for reasons discussed above with respect to claim 1.

Independent claims 22 and 25 recite similar subject matter as claim 8, and are also not anticipated by Merten for similar reasons as discussed above with respect to claim 8.

Claims 9-14 depend from and further restrict claim 8, and claim 23 depends from and further restricts claim 22. These claims are also not anticipated by Merten at least by virtue of their dependency.

B.3 Claims 4, 11, and 18

Claim 4 depends from claim 1 and further recites the steps of "determining whether the instruction is within a second contiguous range of instructions", and "generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions". The Examiner refers to statements in Section 2.1.1 of Merten relating to monitoring short intervals of a branch/candidate branch, as disclosing these steps. Appellants respectfully disagree. Again, Merten does not disclose that a determination is made whether "the instruction [that was identified for execution and that was determined to be within a contiguous range of instructions] is within a second contiguous range of instructions", and then "generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions". Merten, at best, discloses profiling frequently executed

branches that "may be part of a potential hot spot", and this is quite different than "determining whether the instruction is within a second contiguous range of instructions, and generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions" as recited in claim 4.

Claim 4, accordingly, is not anticipated by Merten and patentably distinguishes over Merten in its own right as well as by virtue of its dependency.

Claims 11 and 18 depend from claims 8 and 15, respectively, and recite similar subject matter as claim 4. These claims are also not anticipated by Merten in their own right for the reasons discussed above with respect to claim 4.

B.4 Claims 6-7, 13-14 and 20-21

Claim 6 depends from and further restricts claim 1 and recites that the determining step of claim 1 comprises "comparing an address of the instruction to a set of addresses in a set of registers in a processor to determine whether the instruction is in the contiguous range of instructions."

In rejecting claim 6, the Examiner asserts:

> Merten discloses, *The method of claim 1, wherein the determining step comprises: comparing an address of the instruction to a set of addresses in a set of registers in a processor to determine whether the instruction is in the contiguous range of instructions.*, because every branch in the code region is recorded in the Branch Behavior Buffer.

Final Office Action dated November 3, 2006, page 6.

Appellants respectfully disagree. Merten may disclose a "Branch Behavior Buffer", but does not disclose "comparing an address of the instruction to a set of addresses in a set of registers in a processor to determine whether the instruction is in the contiguous range of instructions" as recited in claim 6. Merten does not anticipate claim 6, and claim 6 patentably distinguishes over Merten in its own right as well as by virtue of its dependency.

Claims 13 and 20 recite similar subject matter as claim 6 and are also not anticipated by Merten for similar reasons as discussed above with respect to claim 6.

Claims 7, 14, and 21 depend from and further restrict claims 6, 13, and 20, respectively, and are also not anticipated by Merten, at least by virtue of their dependency.

Therefore, claims 1-25 are not anticipated by Merten, and it is respectfully requested that the Board reverse the Examiner's Final Rejection of those claims.

DATE: March 5, 2007

Respectfully submitted,

/Gerald H. Glanzman/
Gerald H. Glanzman
Reg. No. 25,035
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method in a data processing system for monitoring execution of instructions, the method comprising:

identifying an instruction for execution;

determining whether the instruction is within a contiguous range of instructions; and

generating execution information relating to the instruction if the instruction is within the contiguous range of instructions.

2. The method of claim 1 wherein the generating step comprises counting each event associated with execution of the instruction if the instruction is within the contiguous range of instructions.

3. The method of claim 2, wherein the counting step comprises:

sending a signal from an instruction cache to a performance monitor unit; and

the performance monitor unit tracking the counting of each event associated with an execution of the instruction if the instruction is within the contiguous range of instructions.

4. The method of claim 1 further comprising:

determining whether the instruction is within a second contiguous range of instructions; and

generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions.

5. The method of claim 1, wherein the execution information includes at least one of a number of visits to the range of instructions and a number of times the instruction has been executed.

6. The method of claim 1, wherein the determining step comprises:

comparing an address of the instruction to a set of addresses in a set of registers in a processor to determine whether the instruction is in the contiguous range of instructions.

7. The method of claim 6 further comprising:

setting the set of registers using a performance tool.

8. A method in a data processing system for monitoring access to data in memory locations, the method comprising:

identifying an access to data in a memory location;

determining whether the memory location is within a contiguous range of memory locations; and

generating information relating to the memory location if the memory location is within the contiguous range of memory locations.

9. The method of claim 8 wherein the generating step comprises:

counting each event associated with access of the memory location if the memory location is within the contiguous range of memory locations.

10. The method of claim 9, wherein the counting step comprises:

sending a signal from a data cache to a performance monitor unit; and

the performance monitoring unit tracking the counting of each event associated with an access of the memory location if the memory location is within the contiguous range of memory locations.

11. The method of claim 8 further comprising:

determining whether the memory location is within a second contiguous range of memory locations; and

generating the information relating to the memory location if the instruction is within the second contiguous range of memory locations.

12. The method of claim 8, wherein the execution information includes at least one of a number of visits to the range of memory locations and a number of times the memory location has been accessed.

13. The method of claim 8, wherein the determining step comprises:

comparing an address of the memory location to a set of addresses in a set of registers in a processor to determine whether the memory location is in the contiguous range of memory locations.

14. The method of claim 13 further comprising:

setting the set of registers using a performance tool.

15. A data processing system for monitoring execution of instructions, the data processing system comprising:

identifying means for identifying an instruction for execution;

determining means for determining whether the instruction is within a contiguous range of instructions; and

generating means for generating execution information relating to the instruction if the instruction is within the contiguous range of instructions.

16. The data processing system of claim 15 wherein the generating means comprises:

counting means for counting each event associated with execution of the instruction if the instruction is within the contiguous range of instructions.

17. The data processing system of claim 16, wherein the counting means comprises:

sending means for sending a signal from an instruction cache to a performance monitor unit; and

the performance monitor unit comprising tracking means for tracking the counting of each event associated with an execution of the instruction if the instruction is within the contiguous range of instructions.

18. The data processing system of claim 15, wherein the determining means is a first determining means and the generating means is a first generating means and further comprising:

second determining means for determining whether the instruction is within a second contiguous range of instructions; and

second generating means for generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions.

19. The data processing system of claim 15, wherein the execution information includes at least one of a number of visits to the range of instructions and a number of times the instruction has been executed.

20. The data processing system of claim 15, wherein the determining means comprises:

comparing means for comparing an address of the instruction to a set of addresses in a set of registers in a processor to determine whether the instruction is in the contiguous range of instructions.

21. The data processing system of claim 20 further comprising:

setting means for setting the set of registers using a performance tool.

22. A data processing system for monitoring access to data in memory locations, the data processing system comprising:

identifying means for identifying an access to data in a memory location;

determining means for determining whether the memory location is within a contiguous range of memory locations; and

generating means for generating information relating to the memory location if the memory location is within the contiguous range of memory locations.

23. The data processing system of claim 22 wherein the generating means comprises:

counting means for counting each event associated with access of the memory location if the memory location is within the contiguous range of memory locations.

24. A computer program product in a computer readable medium for monitoring execution of instructions, the computer program product comprising:

first instructions for identifying an instruction for execution;

second instructions for determining whether the instruction is within a contiguous range of instructions; and

third instructions for generating execution information relating to the instruction if the instruction is within the contiguous range of instructions.

25. A computer program product in a computer readable medium for monitoring access to data in memory locations, the computer program product comprising:

first instructions for identifying an access to data in a memory location;

second instructions for determining whether the memory location is within a contiguous range of memory locations; and

third instructions for generating information relating to the memory location if the memory location is within the contiguous range of memory locations.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.